

**SWORD**

XD-LAB-IMG-010

# Lab10: 图像处理算法模块实 验 2: 伽马矫正

Joseph Xu


2018-7-2

## 修改记录

版本号.	作者	描述	修改日期
1.0	Joseph Xu	初稿	2018-4-1

## 审核记录

姓名	职务	签字	日期

	标题	文档编号	版本	页
	Lab10：算法模块实验 2	XD-LAB-IMG-010	1.0	1 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

## 目录

修改记录.....	1
审核记录.....	1
1. 实验简介.....	5
1.1 概述.....	5
1.2 实验目标.....	5
1.3 实验条件.....	5
1.4 实验原理.....	6
2. PARTA：算法仿真实验流程.....	7
2.1 操作步骤.....	7
3. PARTB：算法模块硬件部署流程.....	13
3.1 操作步骤.....	13
4. 实验结果.....	31




	标题	文档编号	版本	页
	Lab10：算法模块实验 2	XD-LAB-IMG-010	1.0	2 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

图 1-1	算法仿真流程图.....	5
图 1-2	实验连接示意图.....	6
图 3-1	Vivado 下创建新工程.....	13
图 3-2	创建新工程向导窗口.....	13
图 3-3	硬件连接对应位置.....	28
图 3-4	实际硬件连接.....	29

	标题	文档编号	版本	页
	Lab10：算法模块实验 2	XD-LAB-IMG-010	1.0	3 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

## 表目录

未找到图形项目表。

	标题	文档编号	版本	页
	Lab10：算法模块实验 2	XD-LAB-IMG-010	1.0	4 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

## 1. 实验简介

该实验通过 MATLAB 搭建一个伽马校正算法模型，并通过 MATLAB 的 HDL Coder 将模型自动生成为硬件模块，并在 SWORD4.0 上和视频接口集成，实现一个快速的伽马校正算法模块设计。

- 对于初学者，整个实验预计耗时 1.5 小时。
- 对于熟练者，整个实验预计耗时 40 分钟。

## 1.1 概述

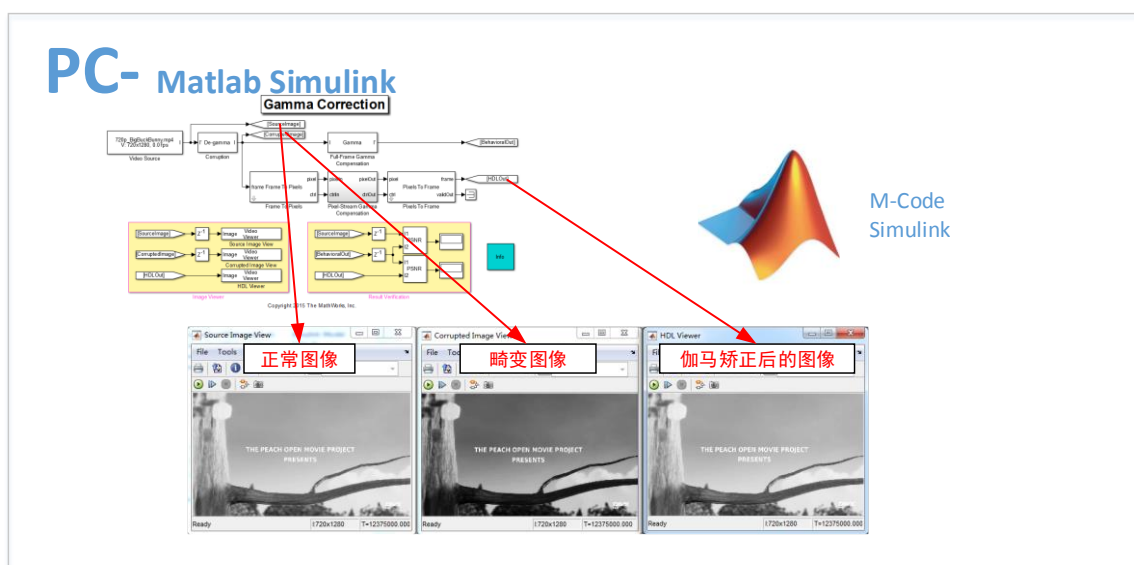



图 1-1 算法仿真流程图

## 1.2 实验目标

本实验的目标为 SWORD4.0 能够正常地在 HDMI 显示器上输出边缘检测的视频图像。

### 1.3 实验条件

类别	名称	数量	说明
硬件	SWORD4.0	1	
	HDMI 信号源	1	如笔记本 HDMI 输出/台式机 HDMI 输出/带 HDMI 输出的 视频机顶盒

	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	5 of 31
	作者	修改日期	公开	
Joseph Xu	2018/7/2			

	带 HDMI 接口的显示器	1	
	HDMI 视频线	2	
	Vivado Design Suite	1	版本：2014.4
	MALTAB	1	版本：R2016a
	视频接口 IP 库	1	FPGA-Image-Library.zip*

\*注：FPGA-Image-Library 为戴天宇开发的一个开源图像处理 IP 库，该 IP 库遵循 LGPL，  
详情请见：<http://fil.dtysky.moe>

1.4 实验原理

该实验的连接方式如下图所示：

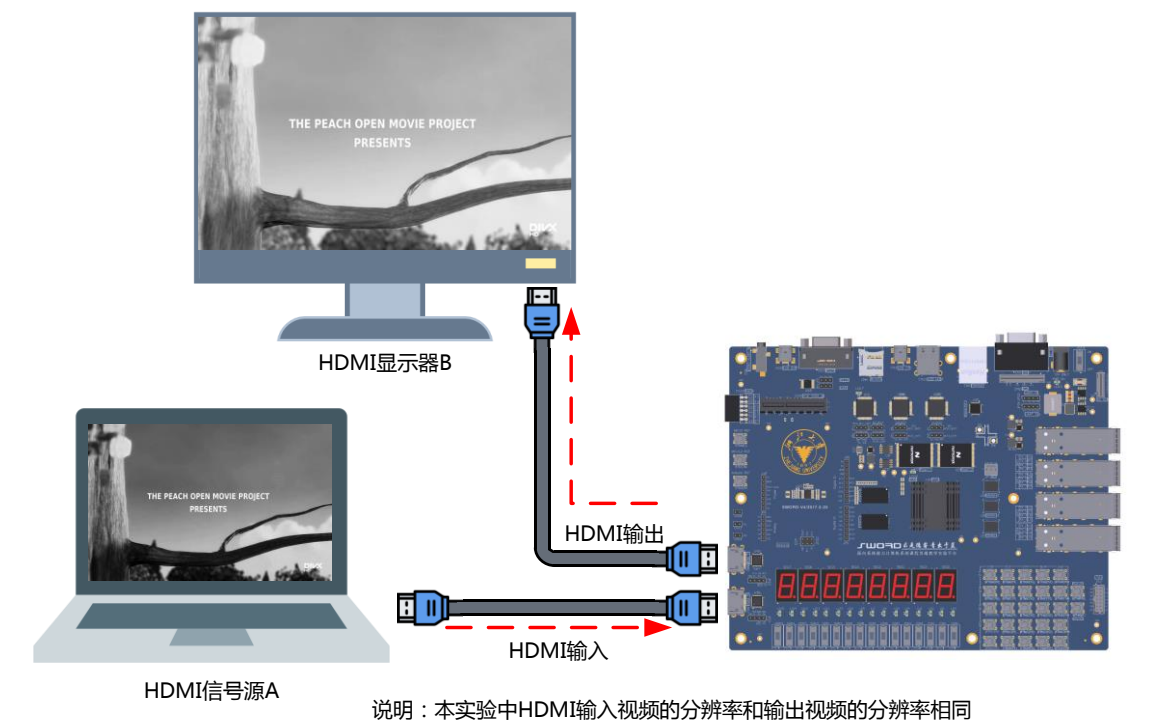


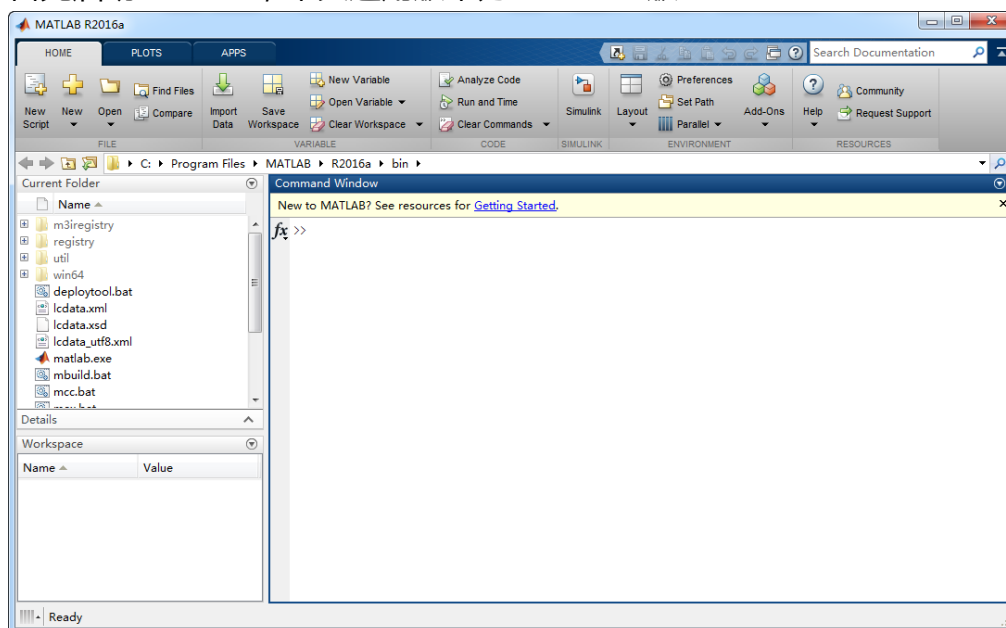
图 1-2 实验连接示意图

## 2. PARTA：算法仿真实验流程

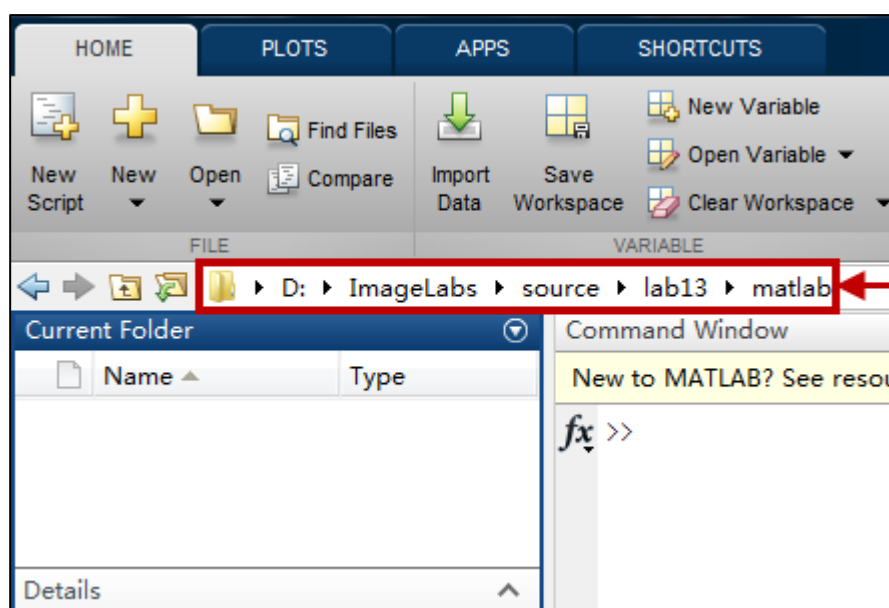
本节将详细描述如何在 MATLAB 的环境下完成实验。请耐心等待，仔细按照图示和文字说明进行操作。


### 2.1 操作步骤

1. 首先启动 MATLAB，本文选用版本为 Windows 版 R2016a。



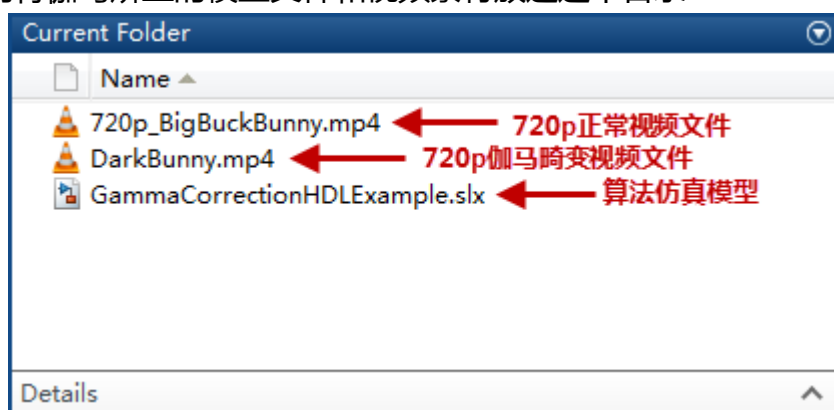
接着我们要设定工作目录，即我们常用来作为存放自己的 MATLAB 文件和图片素材的目录。本文为 D:\ImageLabs\source\lab13\matlab（如果没有自己创建一个）：



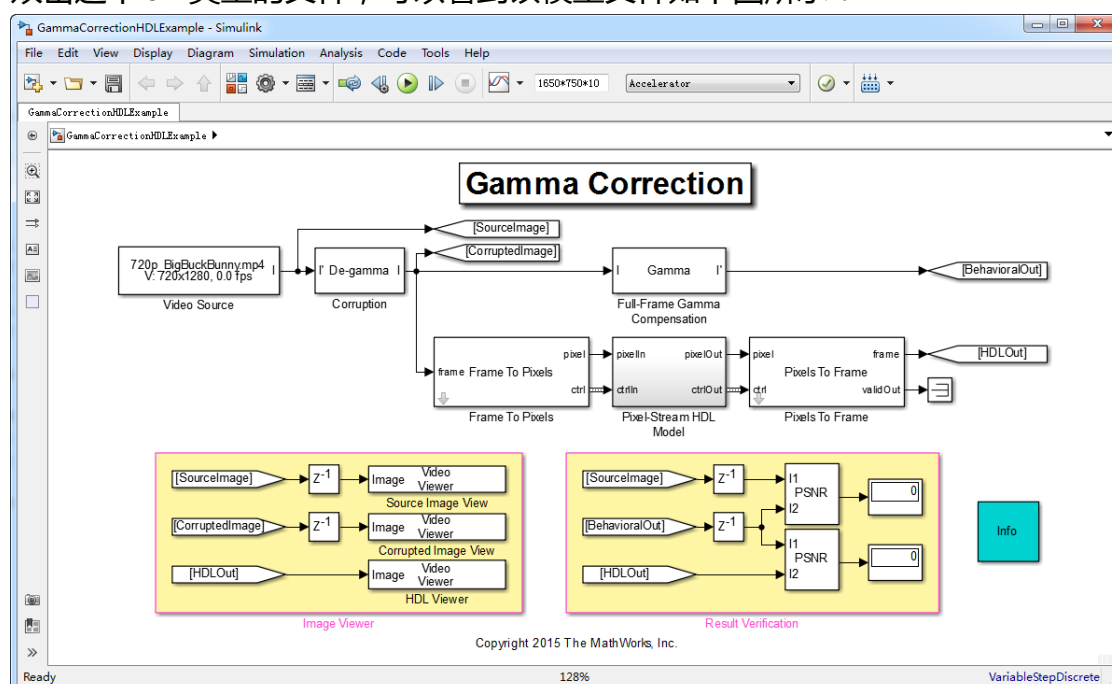
	标题	文档编号	版本	页
	Lab10：算法模块实验 2	XD-LAB-IMG-010	1.0	7 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		



然后将伽马矫正的模型文件和视频素材放进这个目录：




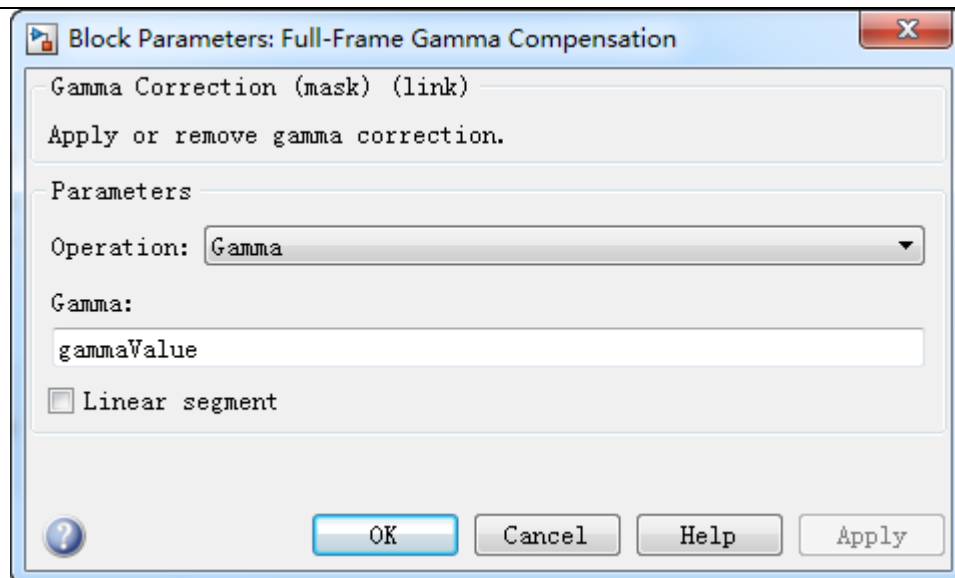
双击这个 slx 类型的文件，可以看到该模型文件如下图所示：



从上图可以看出，该图像算法模型包括了一个视频输入源，其分辨率为 1280x720，帧速为 0.0fps，经过了 2 路处理，其中上面一路为基于完整帧（Full-Frame）的行为模型（即软件处理），下面一路为基于像素流（Pixel-Stream）的 HDL 模型（即硬件处理）。

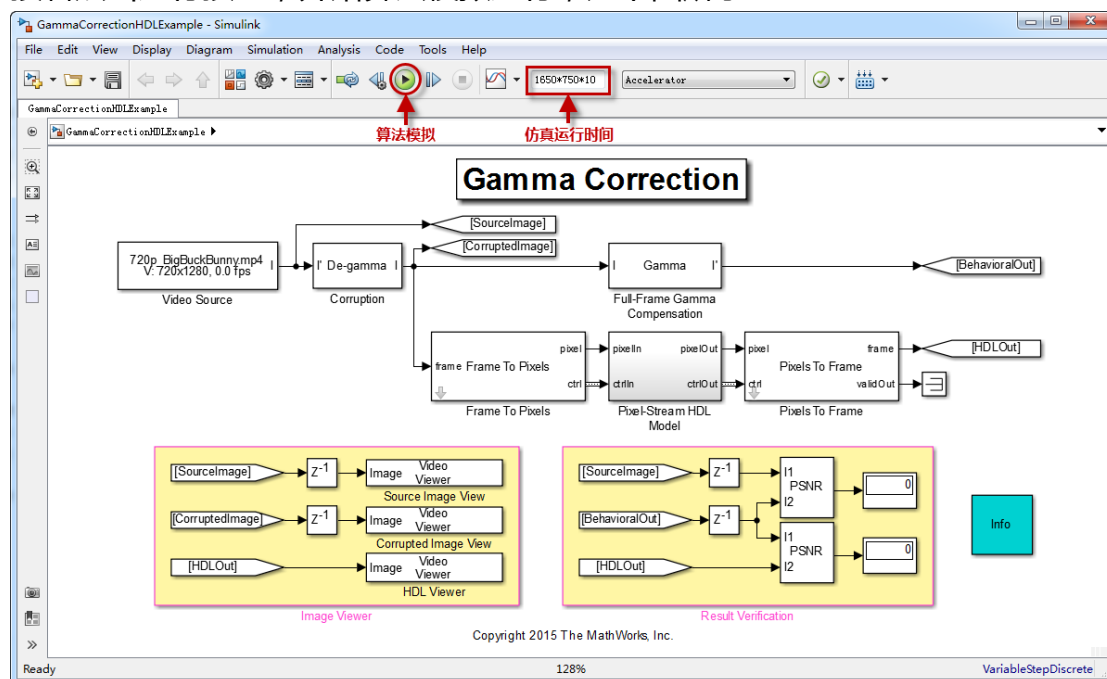
下面我们先来看看 **Full-Frame Gamma Compensation**，双击该模块，如下图所示：

	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	8 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		




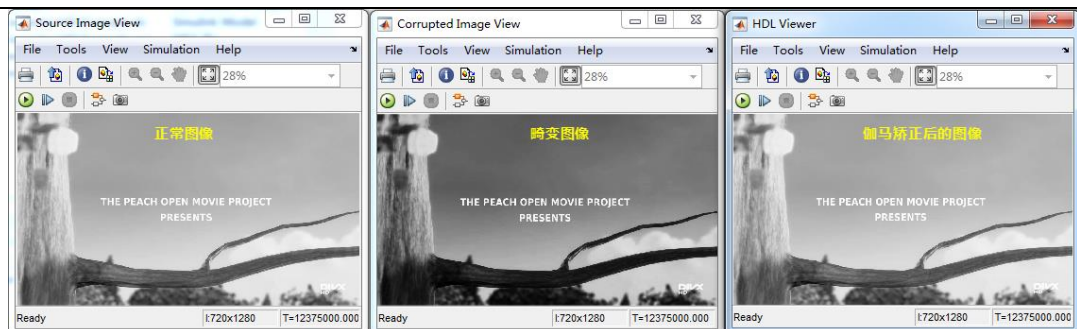
该模型为 Simulink 内置的图像处理工具包 ( Image Processing Toolbox ) 所包含的模块 ( Block ), 所以我们可以看到该模块的参数设置。包括运算的类型为 Gamma , Gamma 运算的值为 gammaValue。关闭对话框。

接着点击运行按钮，开始算法模拟运行，如下图所示：

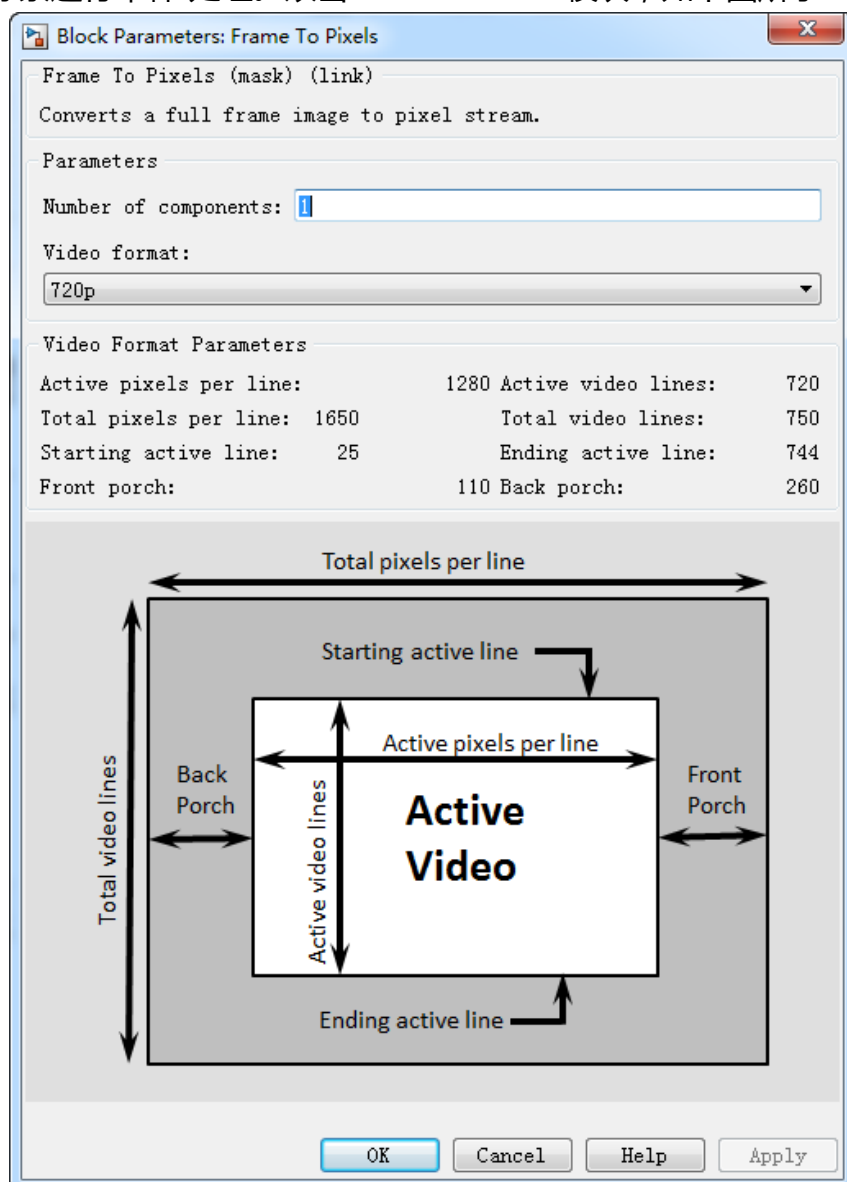


在该模型中，已经内置了算法模拟的运行时间，即  $1650 \times 750 \times 10$ ，即运行 10 帧图像的处理计算。运行后，我们能看到如下结果：


	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	9 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		



接着我们来看看基于像素流的模型，由于该模型后面要转为硬件处理，所以图像数据的处理方式从完整帧转为了像素流，亦即将一幅完整的图像按照一定数量的行像素进行采样处理。双击 **Frame to Pixels** 模块，如下图所示：



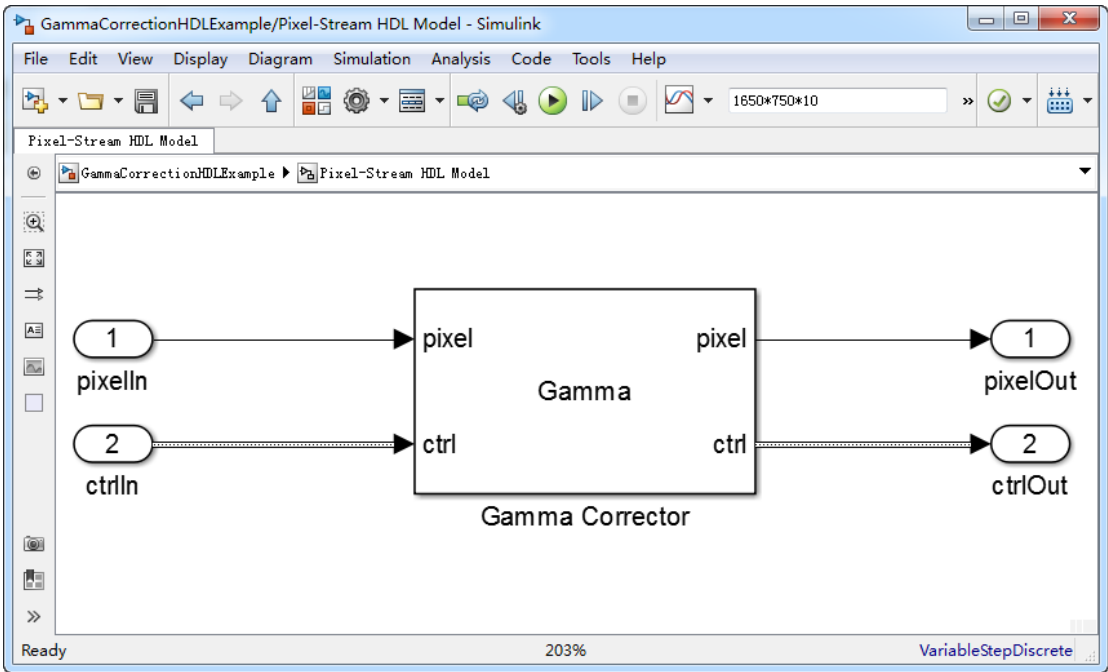
从该模块能很直观的看到一幅视频图像画面的各种说明，包括了：视频分辨率格式，一行包含的像素值与 1 帧包含的行数，以及前沿（Front Porch），后延

	标题	文档编号	版本	页
	Lab10：算法模块实验 2	XD-LAB-IMG-010	1.0	10 of 31
	作者	修改日期	公开	
Joseph Xu		2018/7/2		

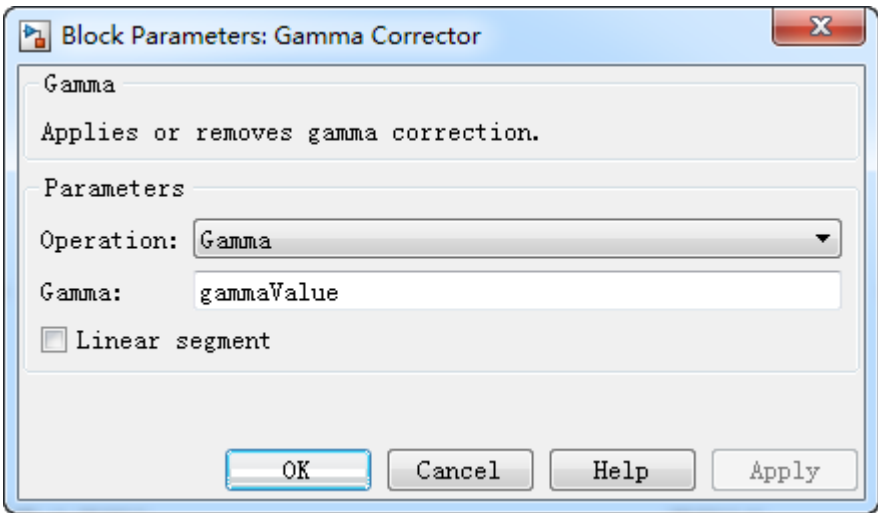
( Back Porch ) 等时序信息。

**注意：这里的设置需要和输入的视频或图像分辨率对应。**

接着双击 **Pixel-Stream HDL Model**，可以看到该模型同之前的类似，是 Simulink 内置的视觉 HDL 工具包 ( Vision HDL Toolbox ) 所包含的模块 ( Block )，如下图所示：



双击 **Gamma Corrector** 模块查看更多细节，如下图所示：



在参数设置上，可以看到该模块的参数设置和之前的一样。

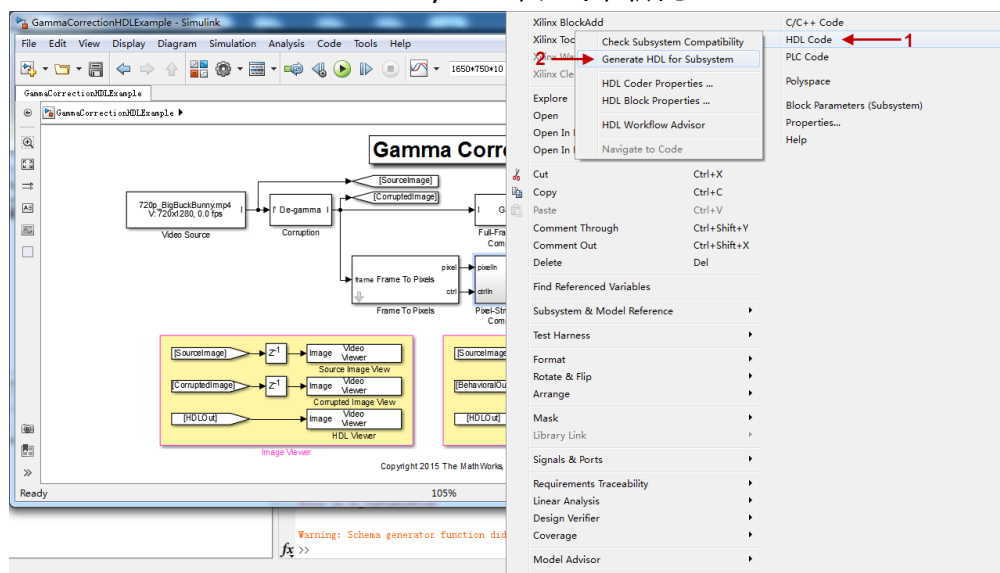
然后我们返回最上层的模型视图，开始验证该算法的硬件处理部分，在 MATLAB 的命令行窗口输入如下命令：

	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	11 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

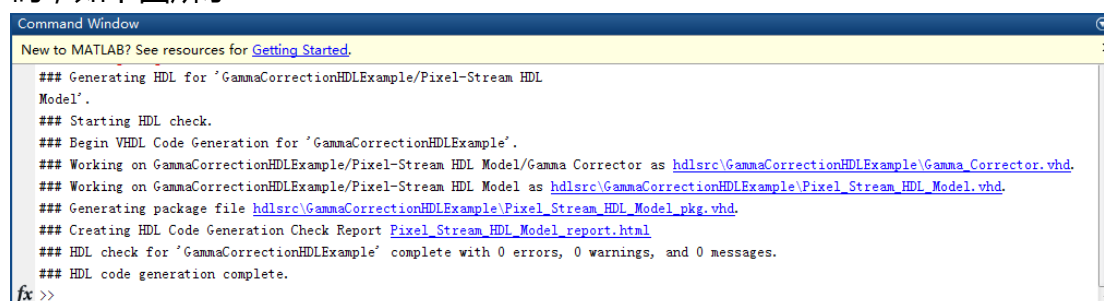
```
gammaValue=2.2
```

上述参数设定了伽马矫正的伽马值，一般来说，该值为 2.2。

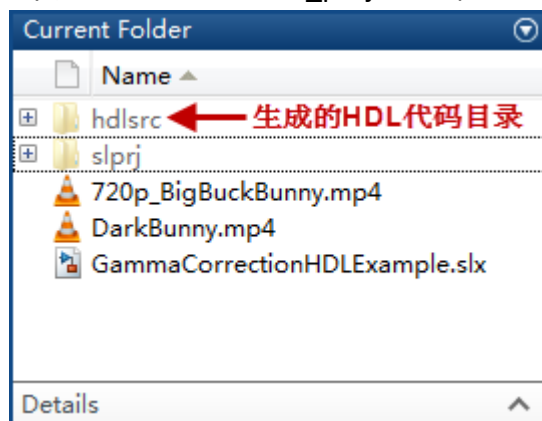
接着我们来生成 HDL 代码，鼠标右键单击 Pixel-Stream HDL Model，然后选择 HDL Code → Generate HDL for Subsystem，如下图所示：




之后我们就能看到 MATLAB 开始生成 Pixel-Stream HDL Model 模型的 HDL 代码，如下图所示：



该 HDL 代码存放于当前工作目录下的 hdl\_proj 目录，如下图所示：



	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	12 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

### 3. PARTB：算法模块硬件部署流程

#### 3.1 操作步骤

前面我们通过 MATLAB 生成了 1 个伽马矫正的算法模型的 HDL 代码，现在我们将该代码部署到 SWORD4.0 上。

1. 首先启动 Vivado 2014.4，然后在主界面点击“Create New Project”，创建工程，如下图所示：




图 3-1 Vivado 下创建新工程

2. 在弹出的向导窗口点击 Next 继续，如下图所示：



图 3-2 创建新工程向导窗口

	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	13 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

3. 接着在窗口页面输入工程名，工程路径和相关选项，按如下信息填写（注意：为保证整个实验的流畅性，请严格按照以下信息填写）：

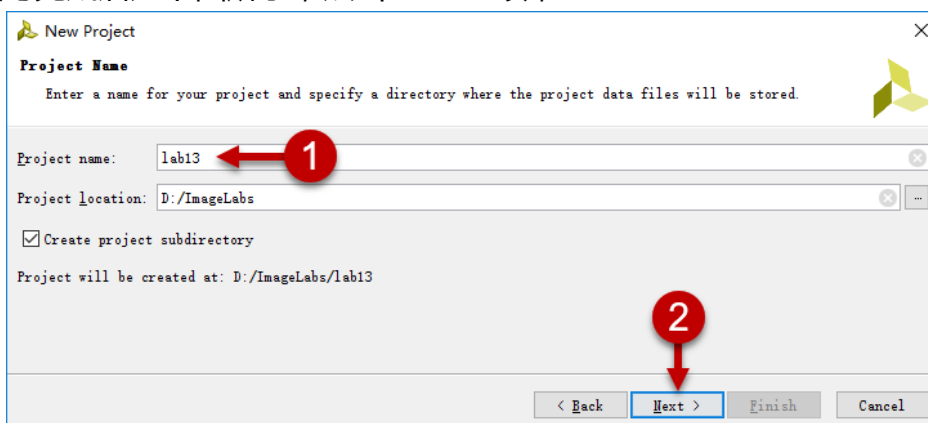
Project name: lab13

Project location: D:/ImageLabs

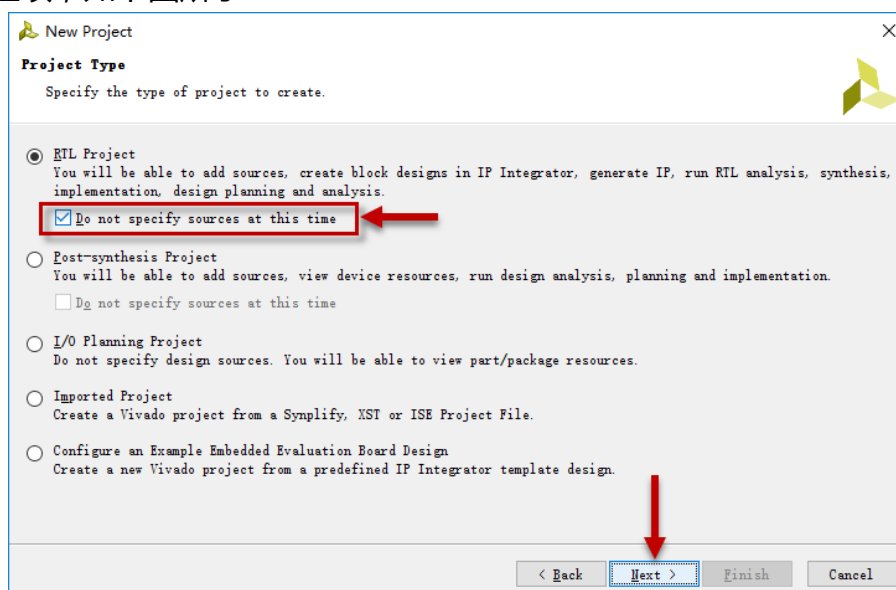
Create project subdirectory: 勾选

**提示：如果本地没有 ImageLabs 这个目录，请自行创建一个**

填写完成后如下图所示，点击 Next 继续；



接着选择工程类型，选择 RTL Project，并勾选 Do not specify sources at this time，点击 Next 继续，如下图所示：



2. 在 Default Part 页面按照如下信息选择目标器件：

Product category: General Purpose


Family: Kintex-7

Sub-Family: Kintex-7

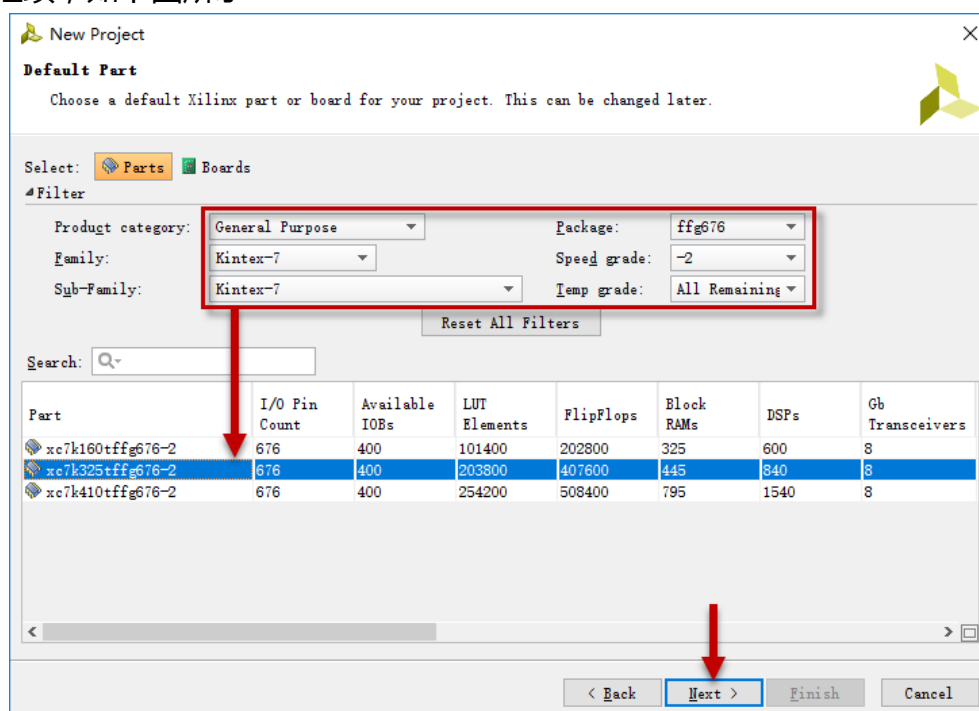
Package: ffg676

Speed grade: -2

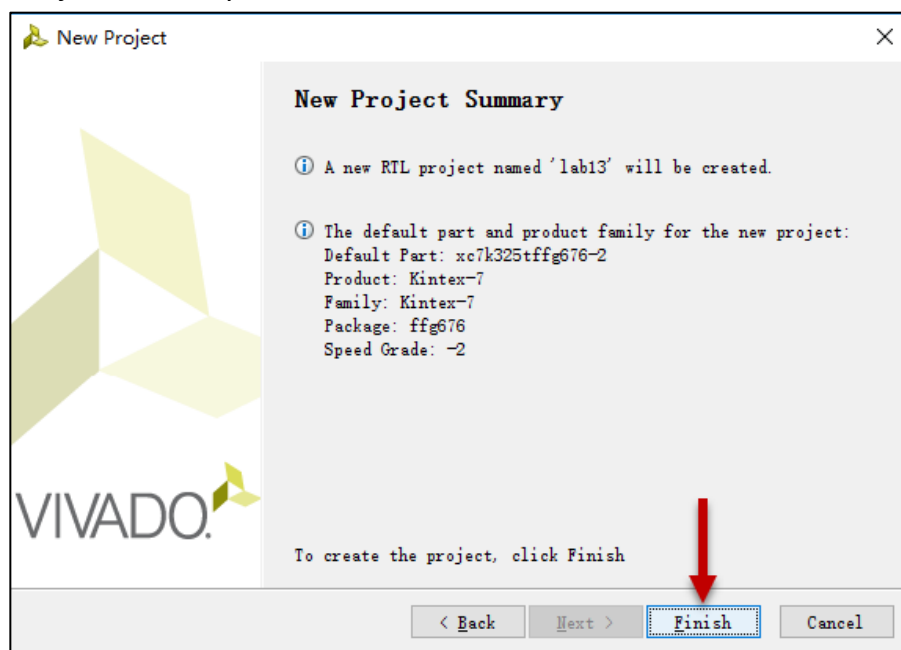
此时在器件列表中剩下的 3 个型号中选择 xc7k325tffg676-2 这个型号，然后点击

	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	14 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		


Next 继续，如下图所示：



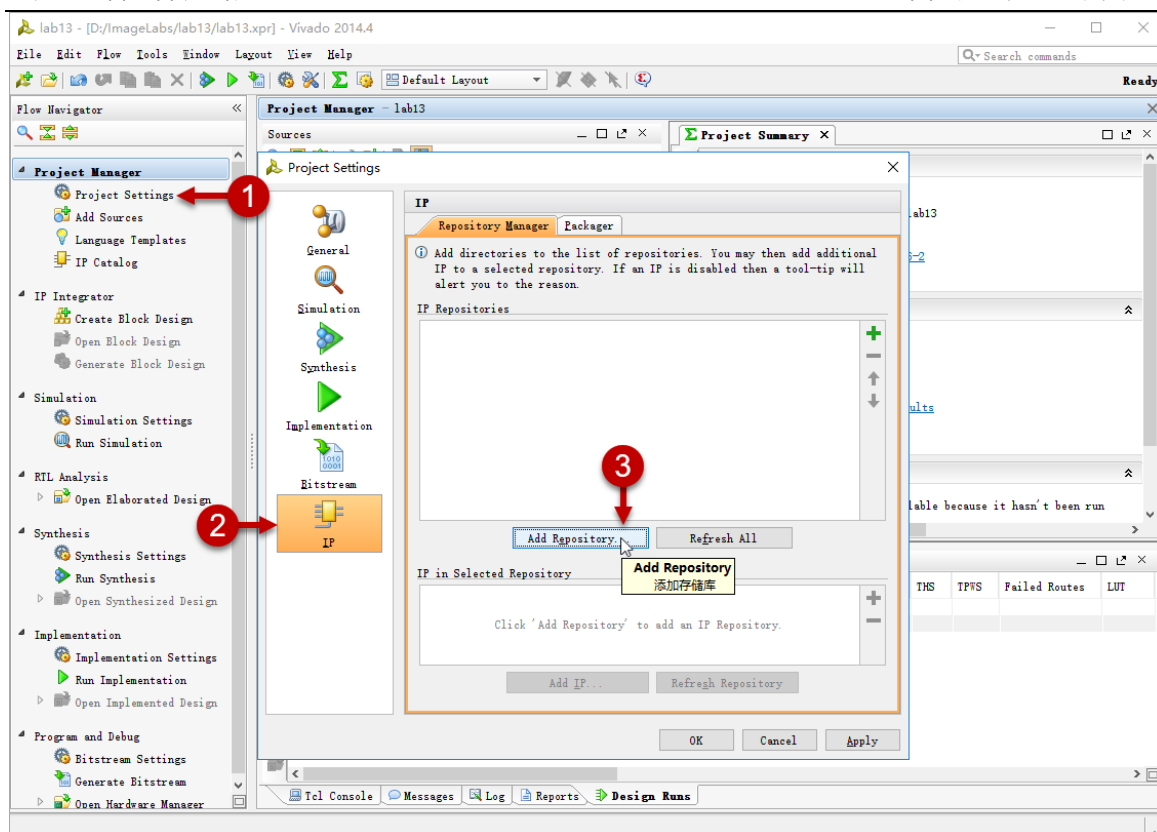
在 New Project Summary 页面直接点击 Finish 完成新工程的创建，如下图所示：



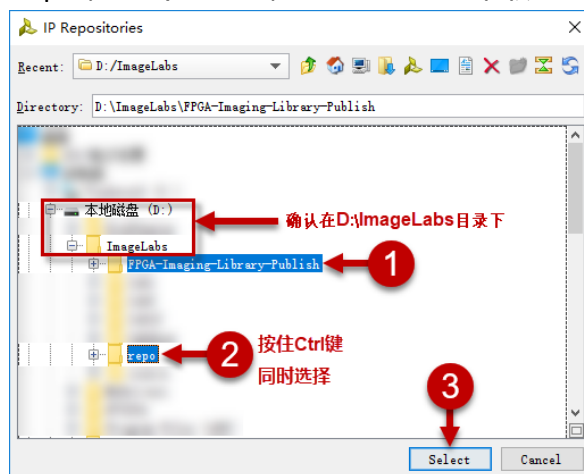
接着要为新的工程添加一个 IP 库 ( repo )，为此我们在 Vivado 主界面的左侧边栏点击 Project Settings，然后在弹出的设置窗口中选择 IP 项，接着点击 Add Repository，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	15 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		




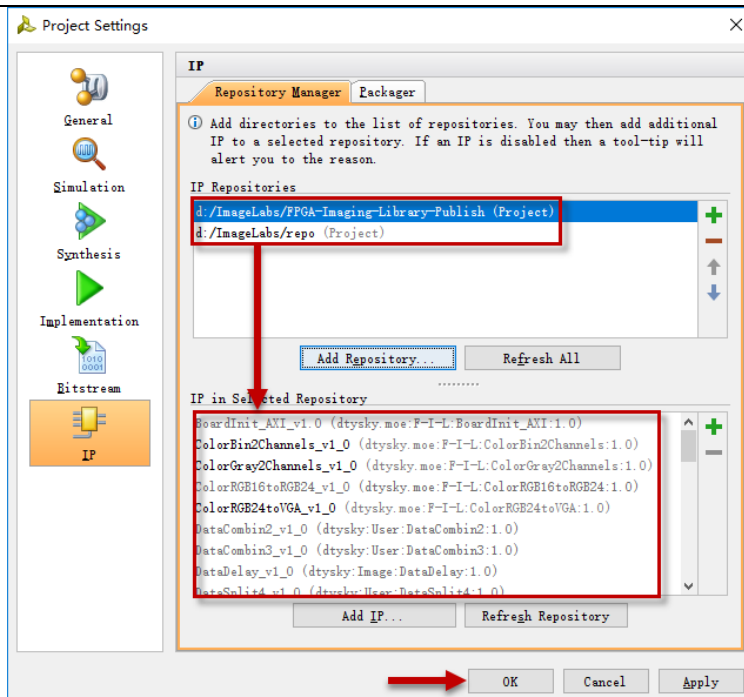


在对话框中找到 D:\ImageLabs 目录，首先选择 FPGA-Image-Library-Publish，然后按住 Ctrl 键，同时选择 repo，点击 Select，整个过程如下图所示：

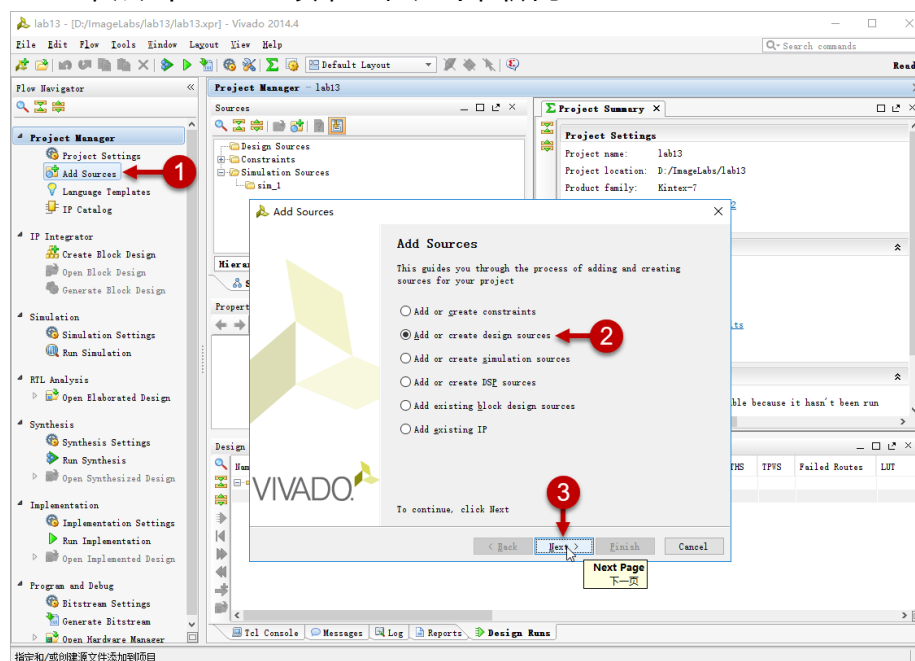


添加好 IP 库后，能看到 Vivado 会自动扫描库中的 IP，如果能看到如下图所示的一些 IP，则表示 IP 库添加成功，此时点击 OK 继续：


	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	16 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

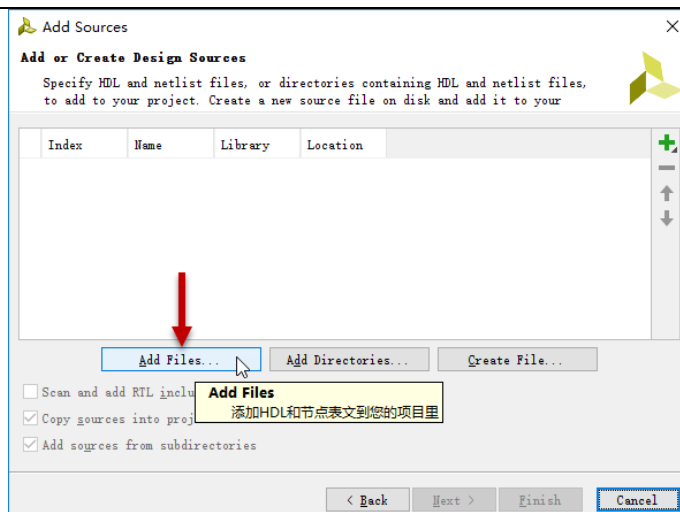


接着在 Vivado 主界面点击 Add Sources 图标，在弹出的窗口中选择 Add or create design sources，点击 Next 继续，过程如下图所示：

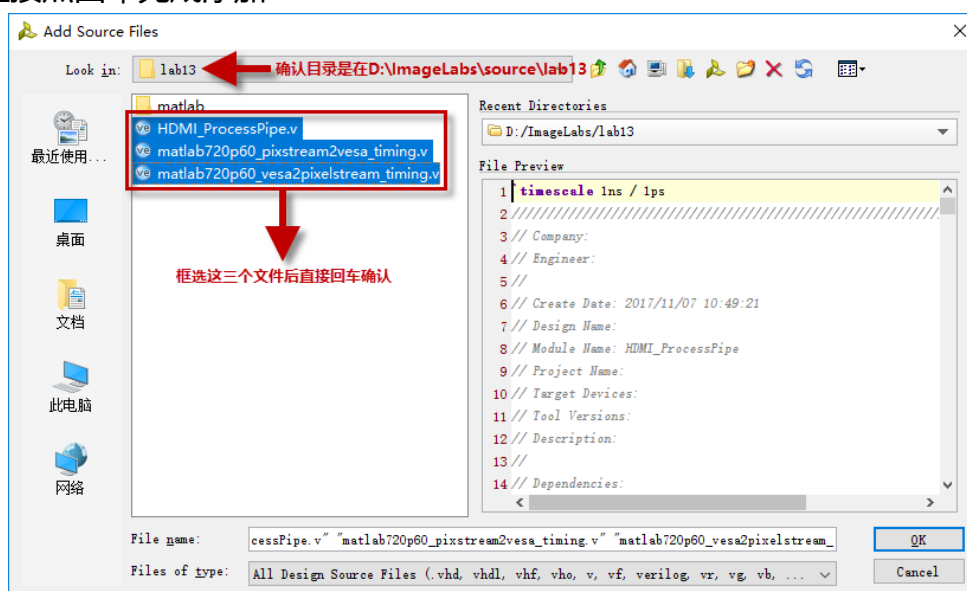


在对话框中点击 Add Files 按钮，如下图所示：


	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	17 of 31
	作者	修改日期	公开	
Joseph Xu		2018/7/2		

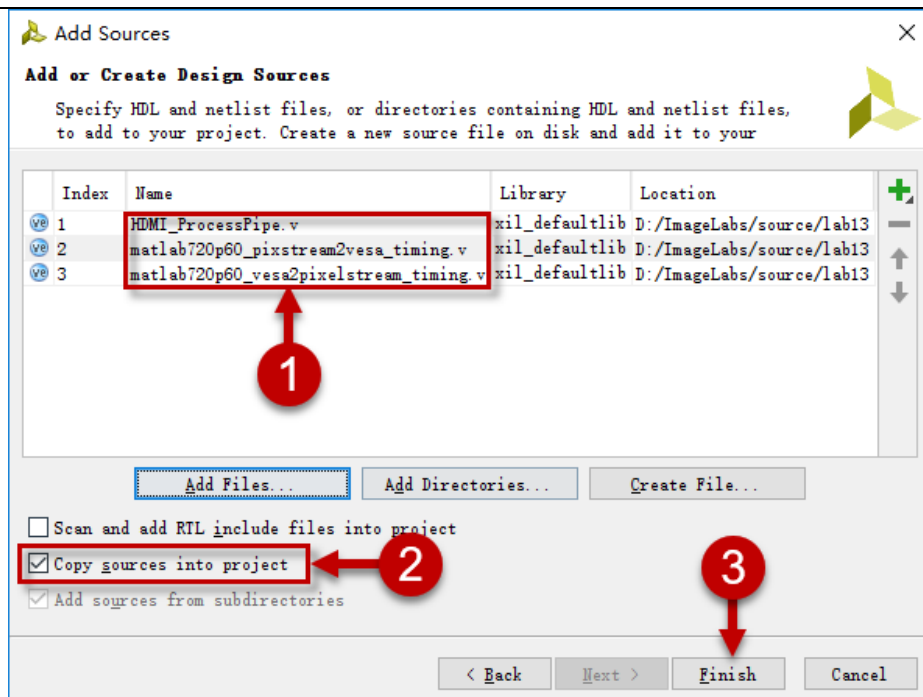


在文件选择窗口，找到 D:\ImageLabs\source\lab13 文件夹，将如图示的 3 个文件选中，直接点回车完成添加：

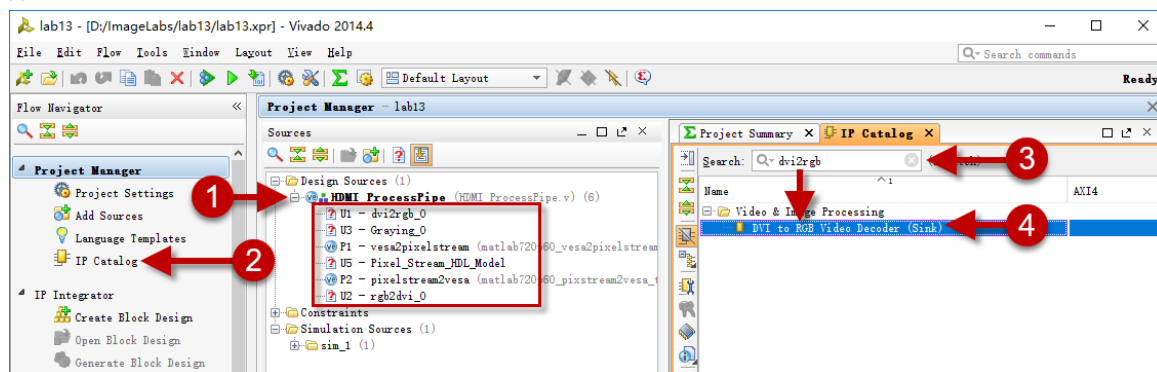


然后在文件添加窗口可以看到 3 个文件被添加，然后勾选 Copy sources into project，点击 Finish 完成文件添加，过程如下图所示：


	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	18 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

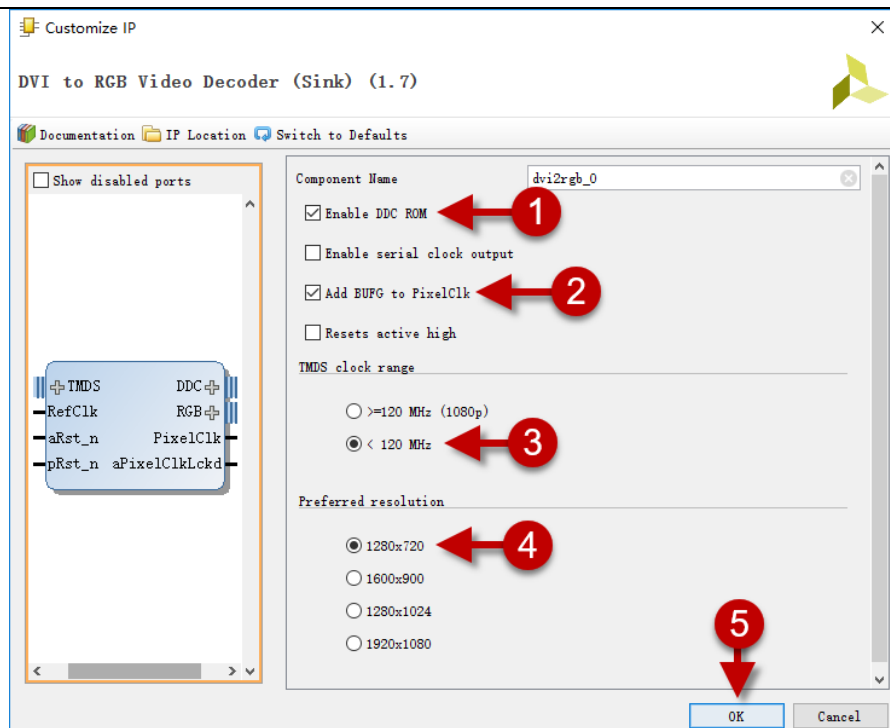


回到 Vivado 主界面，可以看到刚刚添加的是一个顶层设计文件，其中包含了一些 IP 和设计模块，有一些模块前面显示的是带问号的图标，表明该模块或 IP 未添加到工程中。下面我们就来补全，点击 Vivado 主界面的 IP Catalog，然后在弹出的搜索栏中，输入 dvi2rgb，在搜索结果会显示 DVI to RGB Video Decoder，过程如下图所示：

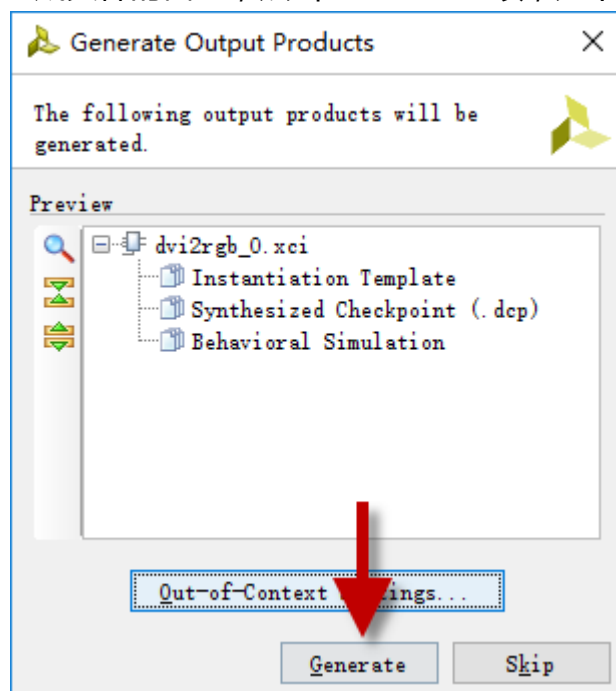


双击这个 IP，Vivado 会弹出该 IP 的配置对话框，按照如下图所示进行配置，并点击 OK 完成：

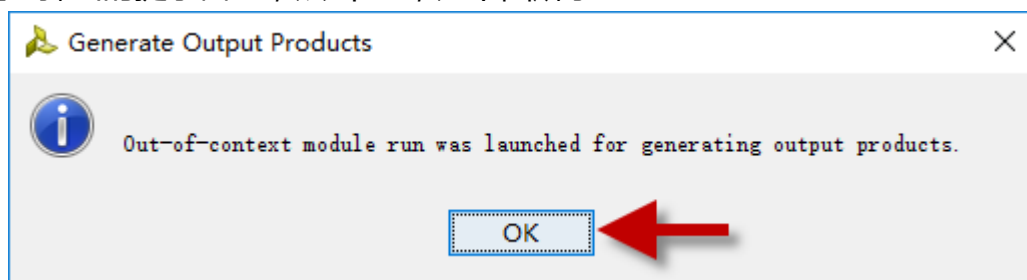
	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	19 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		




接着会弹出一个 IP 生成文件的窗口，点击 Generate 继续，如下图所示：

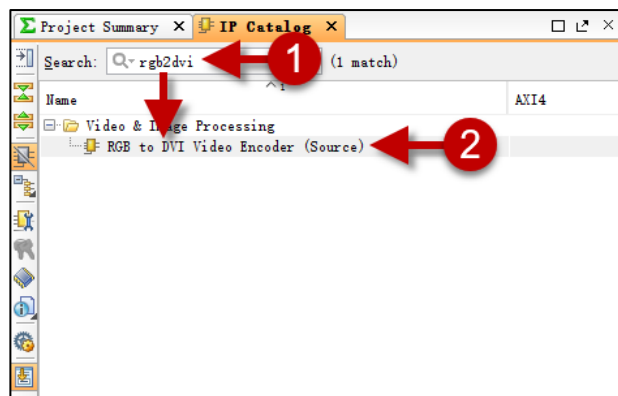


在随之弹出的提示窗口，点击 OK，如下图所示：

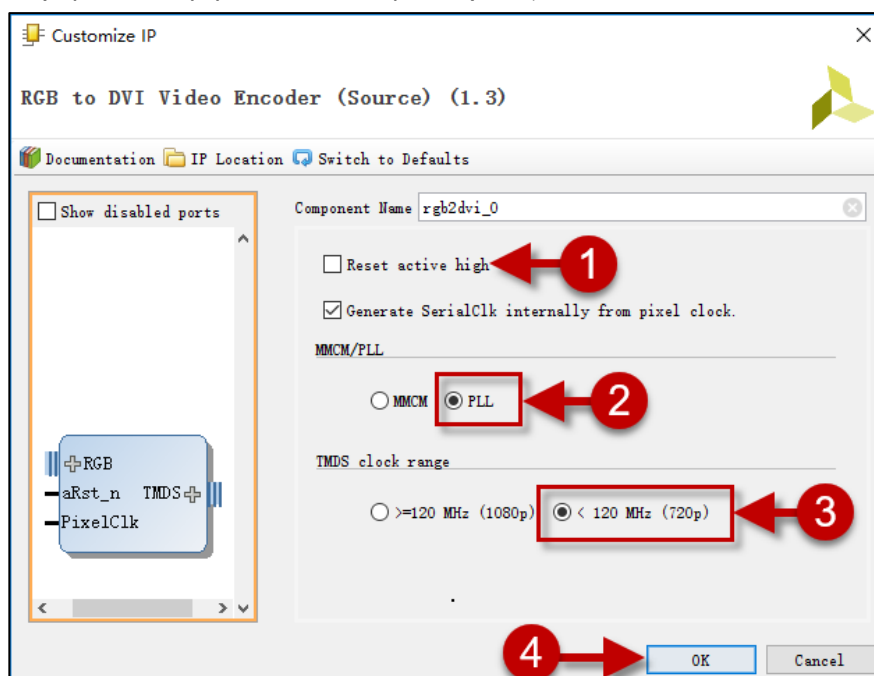


	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	20 of 31
	作者	修改日期	公开	
Joseph Xu		2018/7/2		

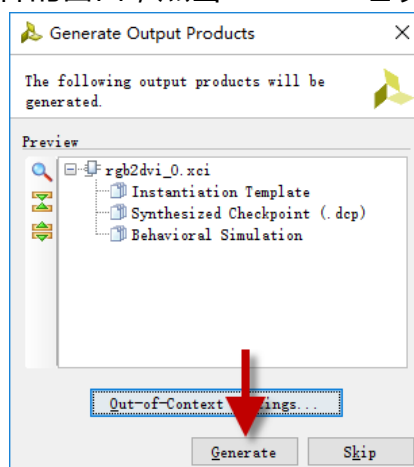
接着按同样的方法，在 IP Catalog 的搜索栏输入 rgb2dvi，并双击搜索结果 RGB to DVI Video Encoder，进行配置，过程如下图所示：




在配置窗口中，按如下图所示进行配置，点击 OK 完成：

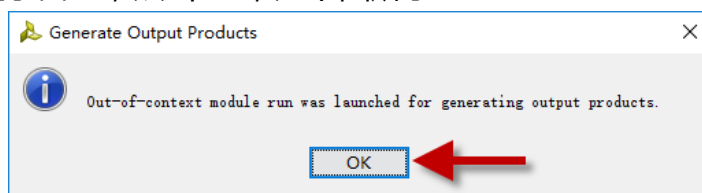


接着会弹出一个 IP 生成文件的窗口，点击 Generate 继续，如下图所示：

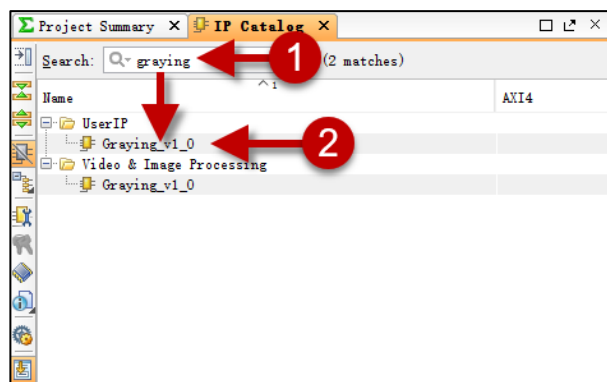


	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	21 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

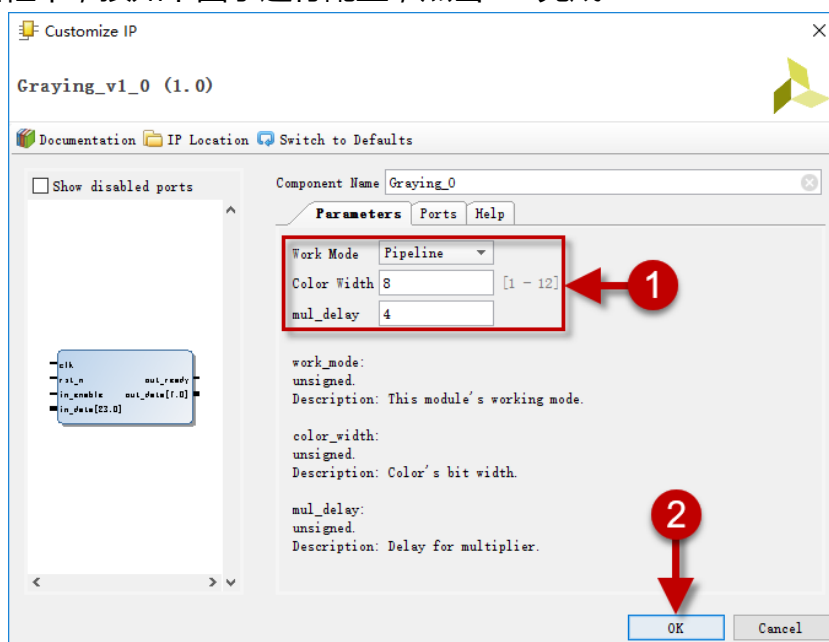
在随之弹出的提示窗口，点击 OK，如下图所示：




在 IP Catalog 的搜索栏输入 Graying，并双击搜索结果，过程如下图所示：

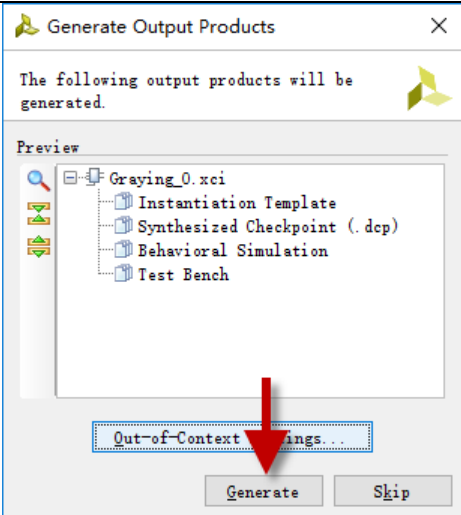


在配置对话框中，按如下图所示进行配置，点击 OK 完成：

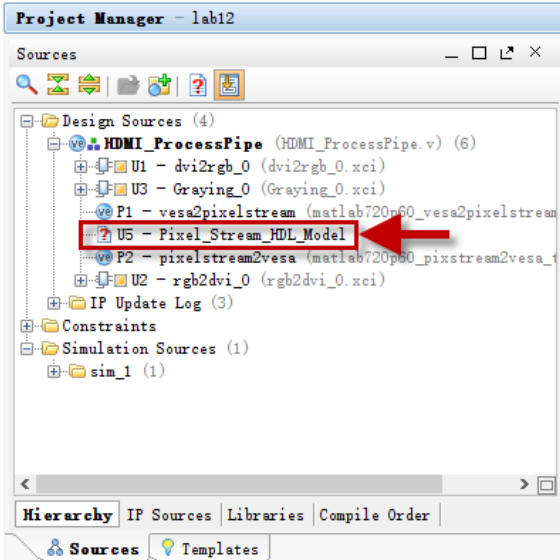


接着会弹出一个 IP 生成文件的窗口，点击 Generate 继续，如下图所示：

	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	22 of 31
	作者	修改日期	公开	
Joseph Xu		2018/7/2		

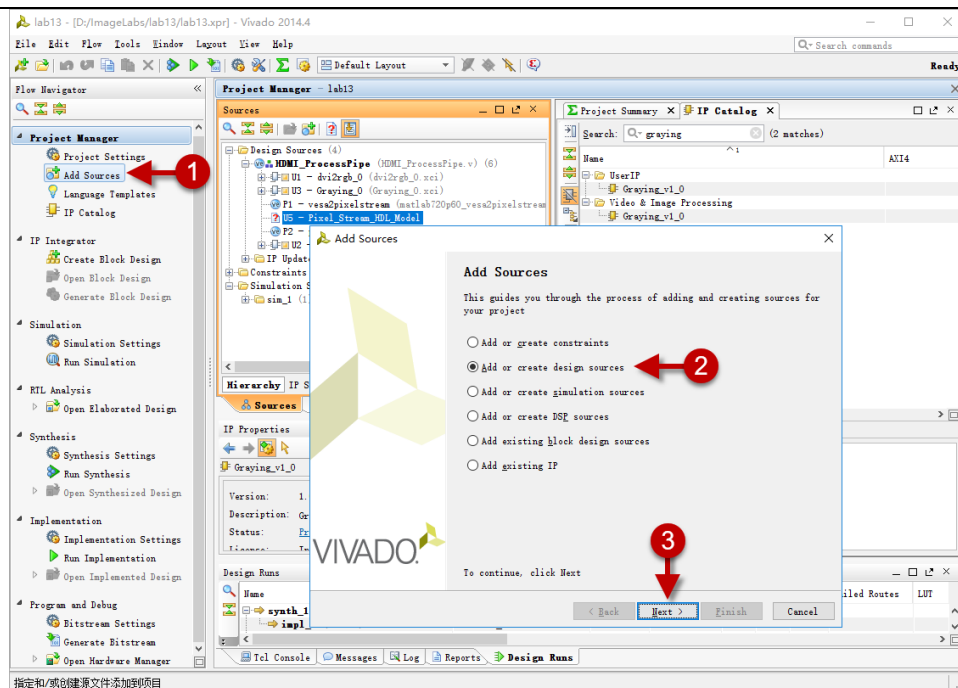


至此，我们已经完成了整个设计的大部分文件导入或添加，但还有一个模块是空着的：Pixel\_Stream\_HDL\_Model，如下图所示：

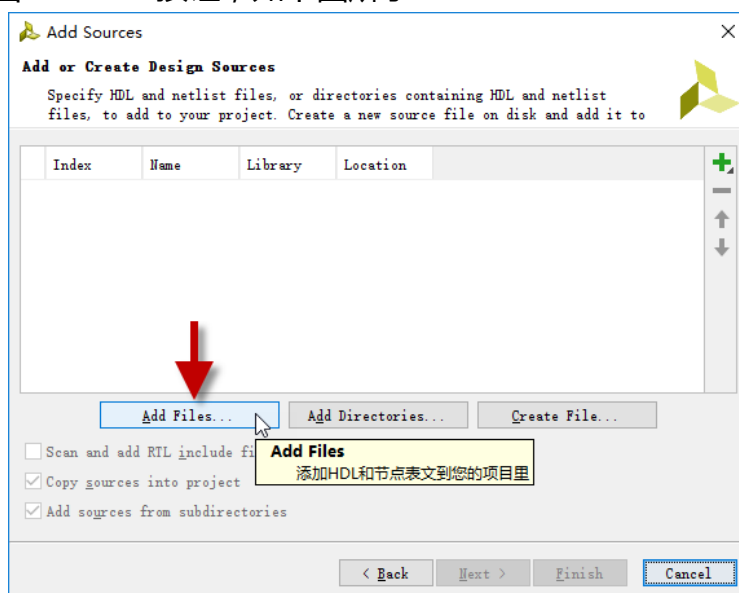


下面我们就将之前在 MATLAB 中仿真的算法模型生成的 HDL 模块代码添加进来，在 Vivado 主界面点击 Add Sources 图标，在弹出的窗口中选择 Add or create design sources，点击 Next 继续，过程如下图所示：



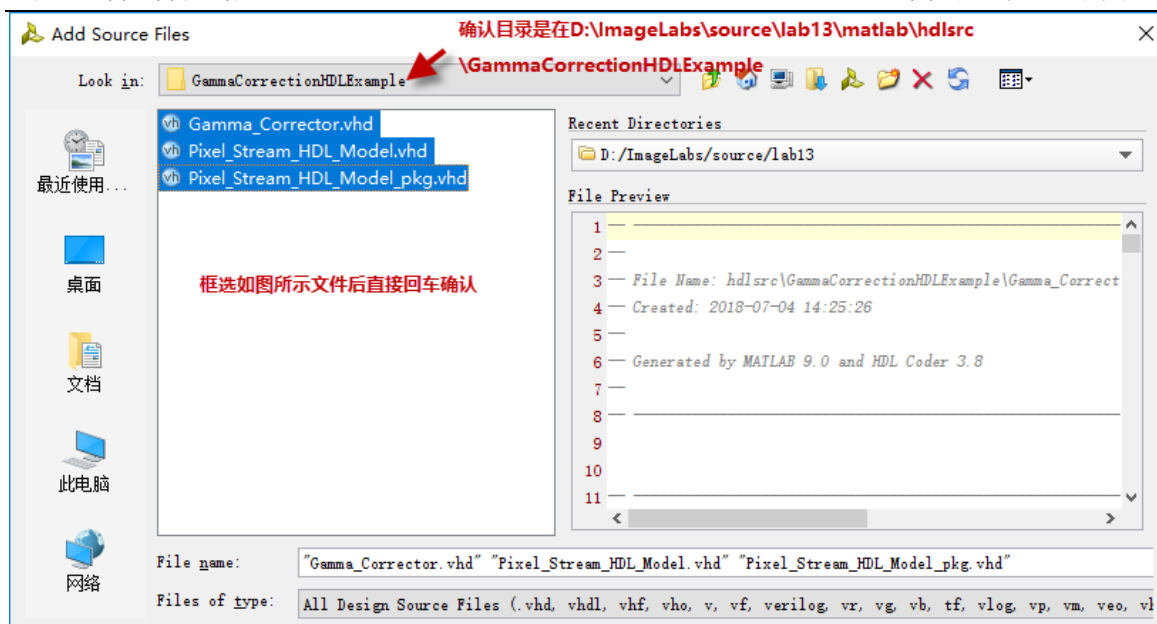


在对话框中点击 Add Files 按钮，如下图所示：

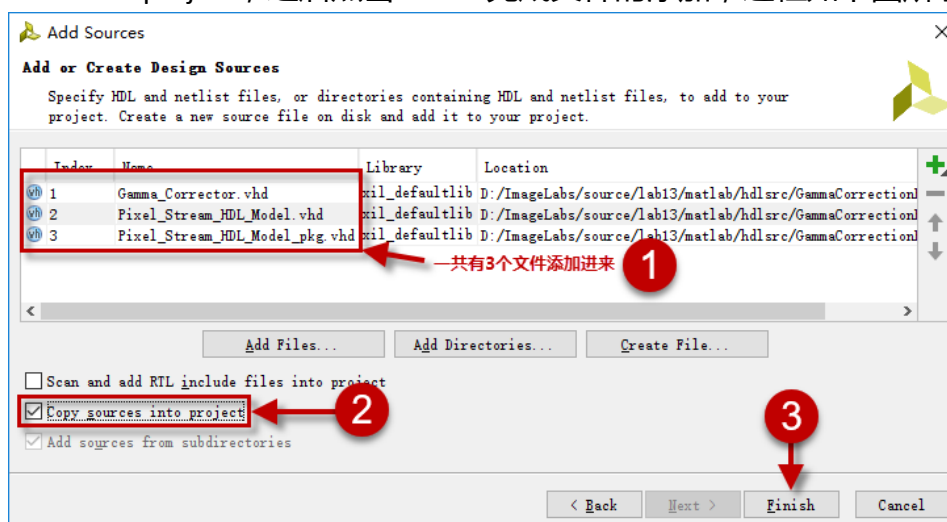


在文件选择窗口，找到 MATLAB 生成的 HDL 代码所在的文件夹，即：  
D:\ImagesLabs\source\matlab\lab13\hdlsrc\GammaCorrectionHDLExample  
将如图示的文件选中，直接点回车完成添加：


	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	24 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

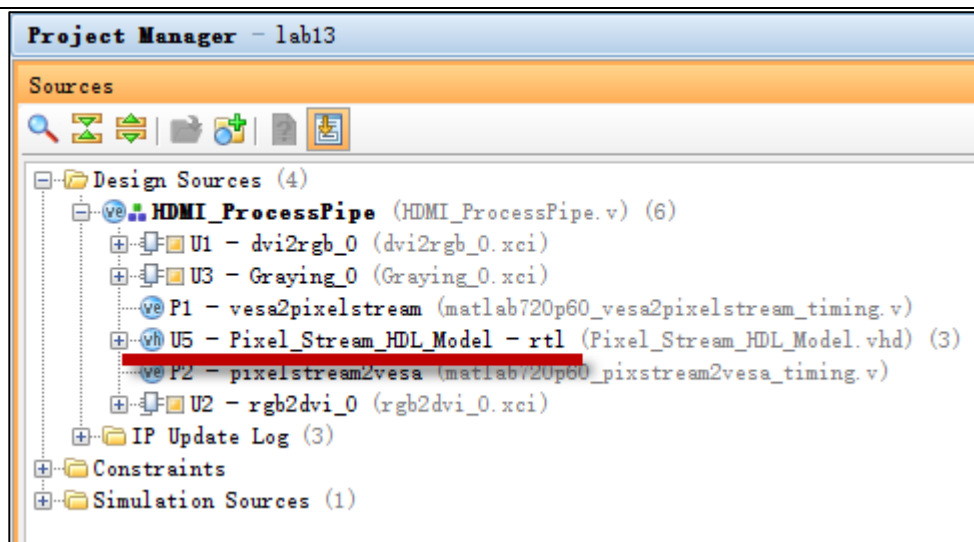


在文件添加窗口，能看到添加的文件，一共有 33 个文件添加进来了，然后勾选 Copy sources into project，之后点击 Finish 完成文件的添加，过程如下图所示：

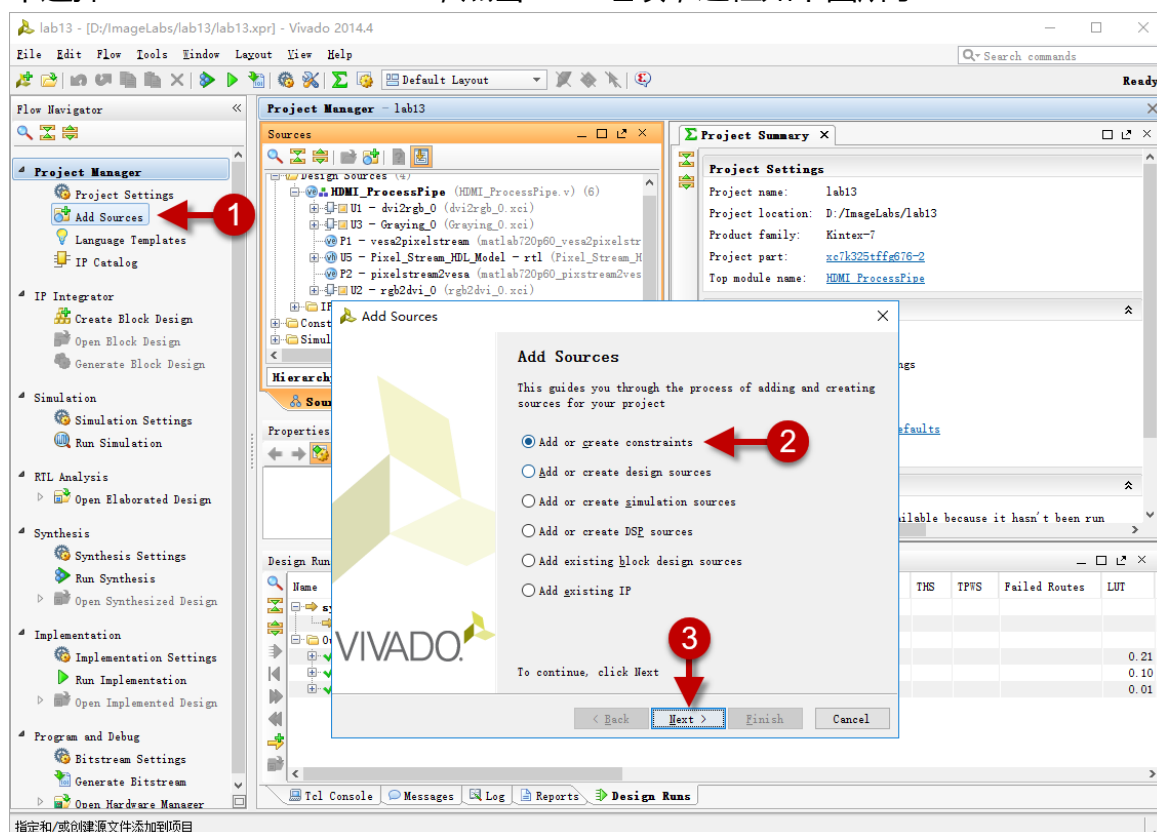


文件添加后，在 Vivado 主界面的 Source 窗口能看到 Pixel\_Stream\_HDL\_Model 也被添加进来了，如下图所示：

	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	25 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		



接着我们要添加约束文件，在 Vivado 主界面点击 Add Sources 图标，在弹出的窗口中选择 Add or create constraints，点击 Next 继续，过程如下图所示：



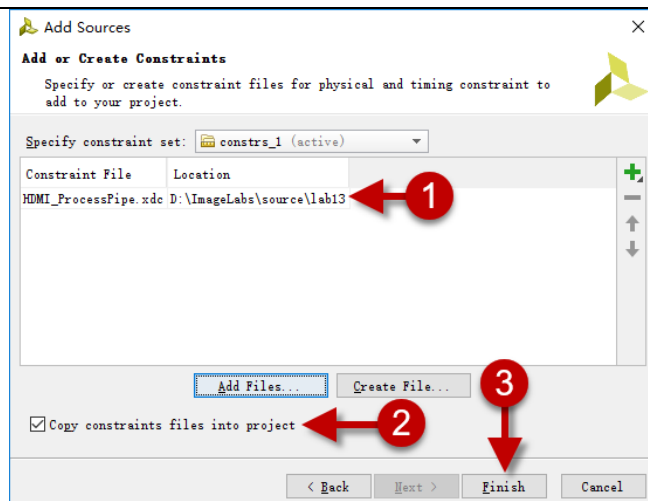
在文件选择窗口，找到约束所在的文件夹，即：

D:\ImageLabs\source\lab13\

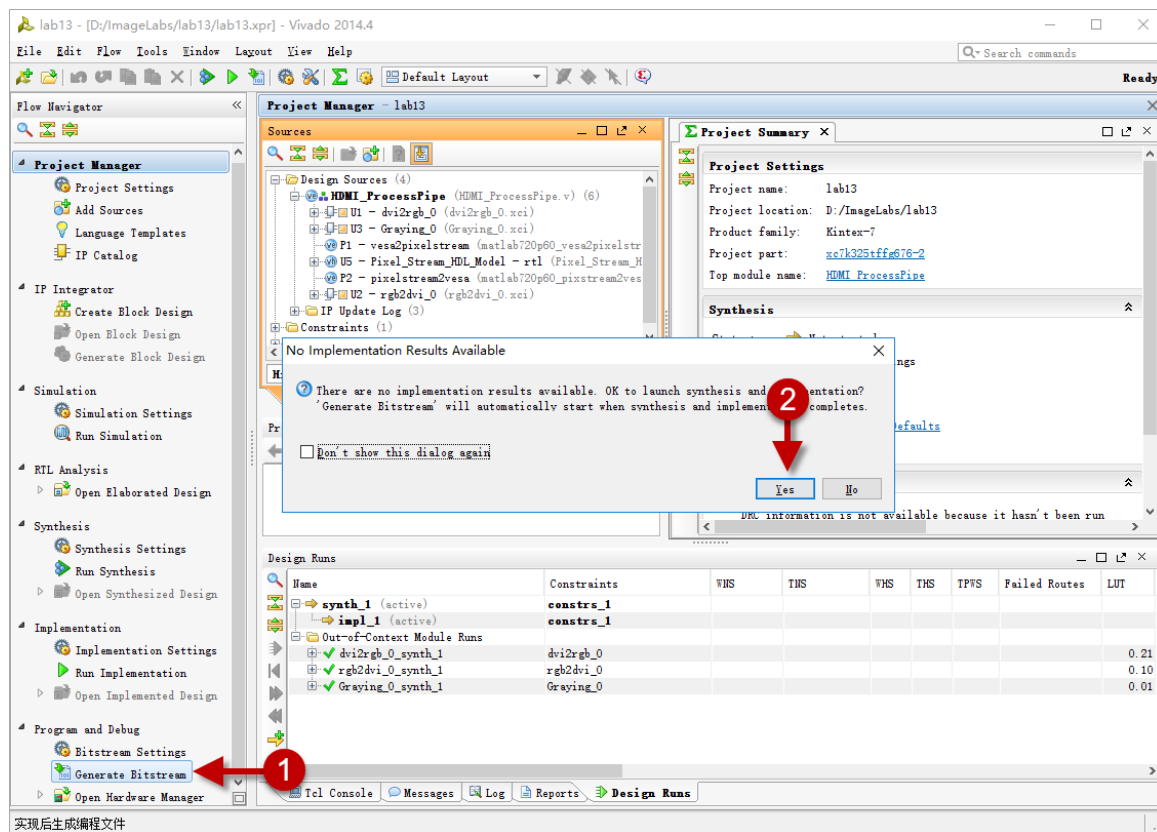
将如图示的文件选中，直接点回车完成添加（**要补图 !!!**）：

在文件添加窗口，检查添加的文件名和文件路径，无误后，勾选 Copy constraints files into project，然后点击 Finish 完成添加，过程如下图所示：

	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	26 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

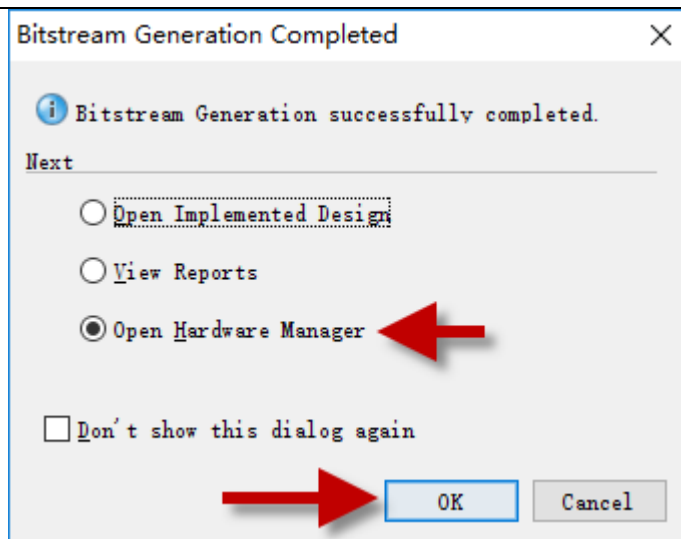


至此，伽马矫正算法模块的 HDL 代码已经部署完毕，在 Vivado 主界面点击 Generate Bitstream，并在随后弹出的提示对话框中点击 Yes 继续，整个过程如下图所示：



大约经过 10 分钟后，Vivado 会弹出 Bitstream Generation Completed 的提示框，表示 bit 文件完成，选择 Open Hardware Manager，然后点击 OK，如下图所示：

	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	27 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		



接着我们需要对 SWORD4.0 硬件平台进行连接，根据下图示意依次进行如下操作：

- 1) 将电源线接上 SWORD4.0，注意此时 SWORD4.0 的开关不要打开；
- 2) 将下载器模块插到 SWORD4.0 的 CN7-JTAG 处，并将下载器的 USB 端口连到电脑；
- 3) 用一根 HDMI 线将 SWORD4.0 和 HDMI 信号源连接上；
- 4) 用一根 HDMI 线将 SWORD4.0 和 HDMI 显示器连接上；
- 5) 打开电源开关

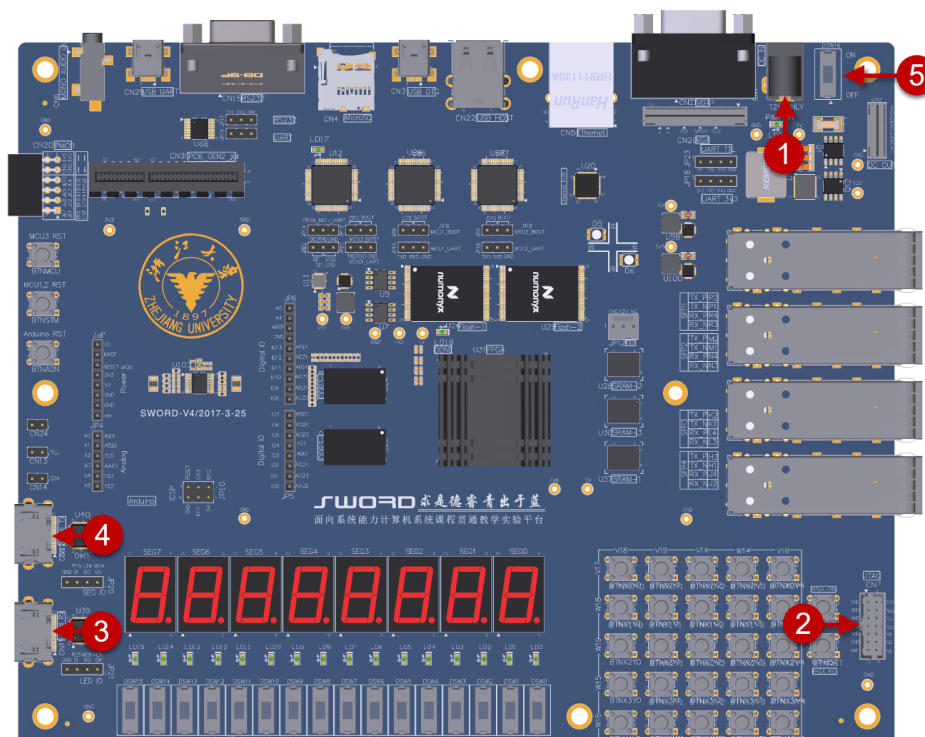



图 3-3 硬件连接对应位置

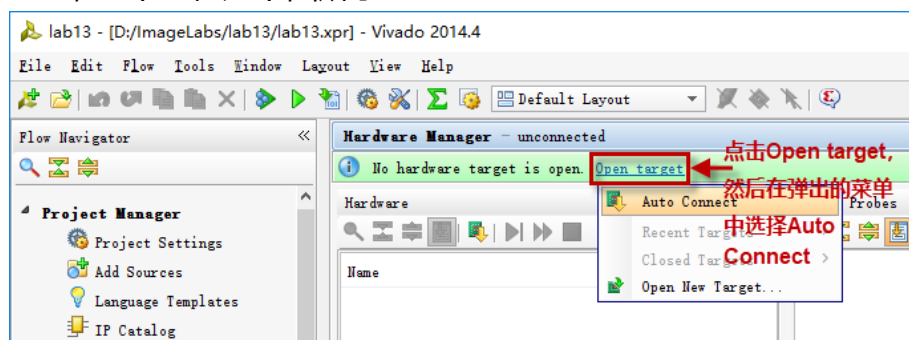
	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	28 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

连接好后的效果如下图所示：




图 3-4 实际硬件连接

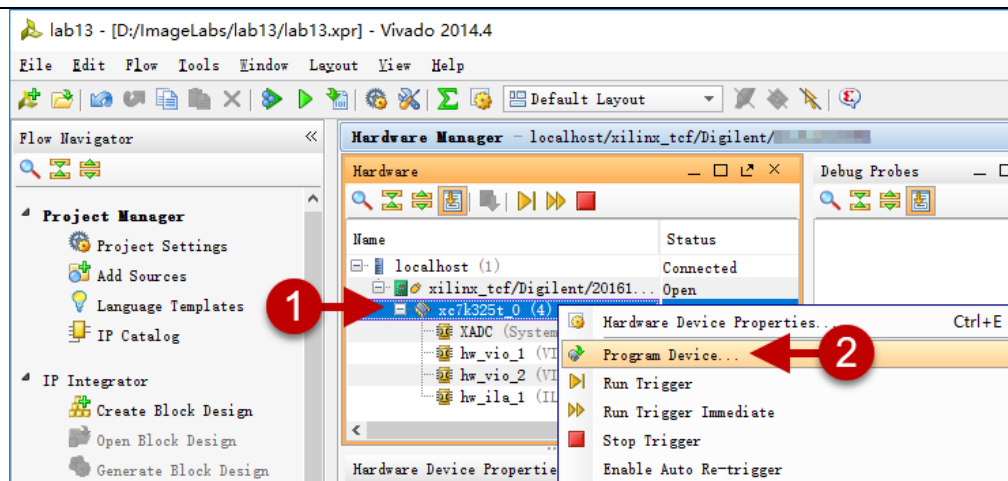
接着在 Hardware Manager 界面下，点击 Open target，在随之弹出的菜单中选择 Auto Connect，整个过程如下图所示：



接着 Hardware Manager 会自动连接下载器并扫描 JTAG，一切正常的话，会显示出扫描到的目标器件：xc7k325t，鼠标右键单击目标器件，在弹出的窗口中选择 Program Device，整个过程如下图所示：

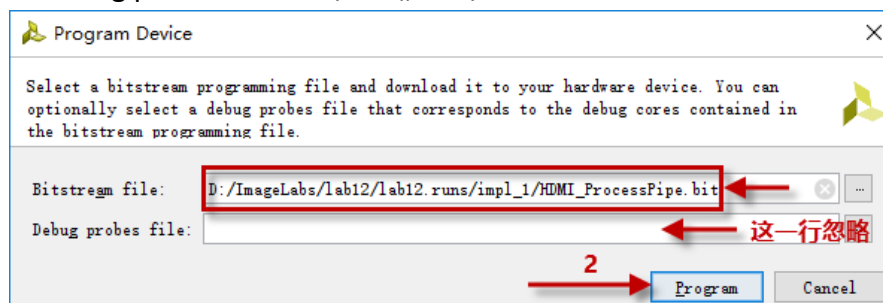
	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	29 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		



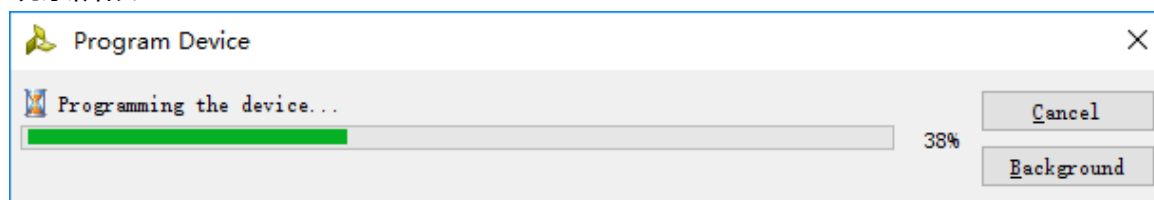



在弹出的对话框中，保持默认设置，直接点击 Program，如下图所示：

提示：如果 Debug probe file 这一栏有输入，可忽略之。



随着如下图所示进度条显示 100%，即表示目标器件烧写完毕。即可进入实验现象观察阶段。



	标题	文档编号	版本	页
	Lab10: 算法模块实验 2	XD-LAB-IMG-010	1.0	30 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		

## 4. 实验结果

此时我们可以将连接 HDMI 输入端口的 HDMI 线在信号源端重新插拔一次，以便让信号源设备重新检测（Detect）一下接收设备，一切正常的话，我们即可在 HDMI 显示器上看到显示画面。



<b>xingdeng</b>	标题	文档编号	版本	页
	Lab10：算法模块实验 2	XD-LAB-IMG-010	1.0	31 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/7/2		